Chapter 14: Configuring Web Access to Logged CRL Entries

This chapter lists tasks that are necessary for creating a web site for your **CRL** entries, in order that collaboration members anywhere in the world can search, view, and annotate your experiment's logbook entries. This web access portion of **CRL** is not required for proper functioning of **CRL**.

Your **CRL** installation should be fully configured and operational before you set up the web portion. Without any special configuring on your part, the application automatically saves a copy of each log entry in HTML format. The web access portion uses the relational data base, the HTML entries, and can be made to use inquiries defined from within **CRL**.

Review section 8.4 *Accessing Archived Entries on the Web* before continuing; you need to envision how the end user will interact with your web interface.

14.1 Editing the Properties File

First make sure the following parameter in the properties file is set properly (see Chapter 15: *CRL's Java Properties*):

```
Logbook.file_location.www_directory
```

14.2 Installing and Configuring Additional Software

Next, install the software you'll need. Go to the web page *CRL - Data Base and Web Access* at `http://www-cpd.fnal.gov/CRL/data_base_web_access.htm`.

For web access, additional software requirements are:

- A webserver of your choice. We have a link to the Apache webserver in case you need one: `http://httpd.apache.org/`
- The tomcat product, used to enable JSPs on the webserver: `http://jakarta.apache.org/`

We provide online instructions for the following tasks:

- Getting, Installing, and Running the Apache Webserver for Linux at `http://www-cpd.fnal.gov/CRL/Apache.html`
- Getting, Installing, and Running Tomcat at `http://www-cpd.fnal.gov/CRL/Tomcat.html`
- Installing the mod_jk Apache to Tomcat plugin at `http://www-cpd.fnal.gov/CRL/Tomcat_Apache_Plugin.html`

14.3 Obtaining and Editing the JavaServer Page

Under the `webapps` directory in the **tomcat** subdirectory of the **CRL** installation, you will find `crlsearch.jsp`, a generic JavaServer page (JSP) that you can modify to do web searches on your **CRL** database. There are some comments in this file which describe simple changes you can make to the web search JSP. To edit this file for more complicated changes and reconfigure it for your experiment, you need to understand JavaServer pages.

14.3.1 Configure Keywords Table

In order to allow users to search on entries by keyword, you must enter the list of keywords into the JSP file. The portion of the file in which to insert this list is shown in the sample below, as is the syntax to use:

```
// vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv <==
// Edit/Add Keywords Below for Your Configuration:  <==
// display_value, value, selected?                  <==
// Keep "KEYWORD_ALL" as FIRST item in list         <==
// ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ <==

Object keywords[][] = {
{KEYWORD_ALL, KEYWORD_ALL, new Boolean(true)},
{"Keyword1", "keyword1", new Boolean(false)},
{"Keyword2", "keyword2", new Boolean(false)},
{"Keyword3", "keyword3", new Boolean(false)},
{"Keyword4", "keyword4", new Boolean(false)},
{"Keyword5", "keyword5", new Boolean(false)},
{NOT+"Keyword1", NOT+"keyword1", new Boolean(false)},
};
```

The last line shows how to define a "NOT" keyword, which can be thought of as the opposite of another keyword in the list (or at least in the database), e.g., NOT_KEYWORD1. This type of keyword makes it possible for users to run a search in which entries linked to a particular keyword can be eliminated. For example, if a user selects NOT_KEYWORD1, then his search will pass over any entry which is linked to KEYWORD1.

14.3.2 Finding Information on JSP

Information about JavaServer Pages dynamically generated web content can be found at the Sun site:

`http://java.sun.com/products/jsp/index.html`

There are lots of other links off of that page. Sun’s Technical Resources page for JSP is located at:

`http://java.sun.com/products/jsp/technical.html`

You can find part of a tutorial that includes JSP information at:

`http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/JSPIntro.html`

A resources page at:

`http://java.sun.com/products/jsp/resources.html`

And a good article:

`http://developer.iplanet.com/viewsource/kuslich_jsp/kuslich_jsp.html`

14.4 Configure "One-Click" Searches

A one-click search (also called a canned search) is a pre-configured search that runs when a user simply clicks a link on a web page. See section 8.4.2 *Run a One-Click Search*. If you plan to create a home page for the web access portion of your experiment’s **CRL** installation, you might want to include some links for canned searches on it.

First, define an inquiry within the **CRL**, and save it (review section 8.2 *Inquiries*). This creates two files, one ending in `.inq` (for inquiry; this one can be used over again within **CRL**), and one ending in `.winq` (for web inquiry). Both files get stored in the directory defined by the parameter `Logbook.file_location.inquiries_directory` (see Chapter 15: *CRL’s Java Properties*).

In your HTML file, link to the `.winq` file via an HREF statement. In the general case, your HREF statement must specify:

- the server on which apache and tomcat are installed
- the directory under tomcat containing the JSP file for **CRL**
- the text `Jsp2.jsp?inquiry=`. When apache sees this, it knows to forward this request to tomcat.

- the name of the `.winq` file (If the file is in the default inquiries directory, do not include the path; if it is in a subdirectory of the default, include the subdirectory portion of the path only. Do not include the file extension `.winq` in the specification.)

The HREF statement has this structure:

```
<A HREF="
http://www.myapache.server/
crl_subdirectory_of_tomcat/
Jsp2.jsp?inquiry=
inquiries_subdirectory/
filename
">
Name of Search
</A>
```

For example, using a `.winq` file named `4hours` and no inquiries subdirectories, the HREF statement might look like this:

```
<A HREF="http://www-xyz.fnal.gov/crl/Jsp2.jsp?inquiry=
4hours">Search the last 4 hours</A>
```